Audio Engineering Society

Convention Paper

Presented at the 139th Convention
2015 October 29–November 1 New York, USA

MixViz: A Tool to Visualize Masking in Audio Mixes

Jon Ford[1], Mark Cartwright[1], and Bryan Pardo[1]

[1] *Northwestern University, Evanston, U.S.A.*

Correspondence should be addressed to Jon Ford (jondavidford@gmail.com)

ABSTRACT

This paper presents MixViz, a real-time audio production tool that helps users visually detect and eliminate masking in audio mixes. This work adapts the Glasberg and Moore time-varying Model of Loudness and Partial Loudness to analyze multiple audio tracks for instances of masking. We extend the Glasberg and Moore model to allow it to account for spatial release from masking effects. Each audio track is assigned a hue and visualized in a 2-dimensional display where the horizontal dimension is spatial location (left to right) and the vertical dimension is frequency. Masking between tracks is indicated via a change of color. The user can quickly drag and drop tracks into and out of the mix visualization to observe the effects on masking. This lets the user intuitively see which tracks are masked in which frequency ranges and take action accordingly. This tool has the potential to both make mixing easier for novices and improve the efficiency of expert mixers.

1. INTRODUCTION

Audio mixing is the process of combining many different audio recording (tracks) together into a limited number of channels. Auditory masking is "the process by which the threshold of audibility for one sound is raised by the presence of another (masking) sound." [1] A typical high-level goal many users have when mixing is to make each sound in the mix as clear and distinct as possible. This can be difficult in the case where one or more tracks are masked by other tracks. The resulting mix is often described as "muddy" and the intelligibility of vocals or the clarity of a melody line may be obscured. There are cases where masking is desirable in a mix (e.g. blending together a horn section). However, even in a mix where some masking is desired, it is almost certainly the case that there are also tracks in the mix that the user does not want to be masked.

One commonly-used method to reduce masking is to adjust equalization settings (EQ) of two tracks involved in masking so that their energy does not

overlap strongly in frequency. This involves boosting or cutting specific frequency bands in a sound. A second widely-used method to reduce masking is adjusting panning settings to change the perceived spatial position of a sound in a stereo mix. As two sounds become more spatially separated, masking is reduced [2]. Both methods depend on a clear understanding of which tracks and/or frequencies are coming into conflict.

It is often difficult for the non-expert user to know which tracks to apply these methods to, and in which frequency ranges. They simply know it sounds bad, but they don't know where the interference is occurring. Even if the user can detect and eliminate masking in one track, their actions can cause another track to become masked. Typically, it takes either an expert audio engineer or considerable trial and error to resolve these issues and create a quality mix.

To ease this task, we have created MixViz, a visualization tool that shows the user which tracks are being masked and in which frequency ranges. Mixviz uses the Glasberg and Moore model of loudness and partial loudness [3] to find masking between tracks. We have extended this model to account for spatial release from masking effects, letting us display the effects of both frequency and panning on perceived masking. With MixViz, the user retains total control over the mix and is able to make informed mixing decisions that emphasize or deemphasize masking. MixViz is useful for both novices and experts and provides the user with information previously only obtainable from an expert listener's judgment.

The software is available for free under the GPL v3.0. It can be found at
`http://dx.doi.org/10.5281/zenodo.22203`

The paper outline is as follows. Section 2 overviews related work. Section 3 describes the Glasberg and Moore percpeptual model, including our extensions to account for spatial release of masking. In Section 4 we describe MixViz, the masking visualizer. Section 5 walks the reader through an example scenario, illustrating how MixViz is used. Section 6 describes future work. Section 7 concludes the paper.

2. BACKGROUND

In the past decade, multiple researchers have sought to make mixing more intuitive [4, 5, 6, 7, 8, 9, 10, 11, 12, 13]. One approach to this problem is to remap (and often reduce) the dimensions of the mixing parameter space to more intuitive dimensions such as semantic or perceptual dimensions. For example, researchers have enabled users to control individual tools in the mixing process such as equalizers [4, 5, 6, 7], and reverberators [4, 9] using low-dimensional semantic parameters (e.g. "tinny", "boomy" knobs). In [14], authors developed a mapping of mixing levels that is more in line with perception. In [8], authors developed a low-dimensional mapping of levels and equalization that encourages exploration. While such tools may help users at different stages of the mixing process, none of these tools directly address the difficult problem of auditory masking.

Another approach researchers have taken to make mixing easier is to completely automate the mixing process. This has been accomplished either by adhering to sets of expert-derived rules [15] or by automatically quantifying and minimizing masking [16, 17, 18, 19]. The problem with these approaches is that the user loses artistic control over the mix. If the user wants to emphasize masking on certain tracks (e.g., blending horns) while reducing it on others, they do not have that choice.

MixViz is the first work we are aware of that analyzes audio for masking and still gives the user total control of their mix. We build on the multi-track masking models used in some automatic mixing approaches, however, instead of using this model to automatically reduce masking as in [16, 17, 19], we use it to inform the user where masking is occurring. This is a unique approach that allows the user to make informed mixing decisions without sacrificing creative control.

3. THE MASKING MODEL

While masking is a perceptual phenomenon, hearing scientists have developed models that we can use to predict when masking will occur. In MixViz, we use a custom extension of Glasberg and Moore's time-varying model of loudness and partial loudness [3] (hereafter referred to as the GM model).

The Glasberg and Moore (GM) model calculates two values for an audio signal: loudness and partial loudness. The inputs to the model are the time domain

audio signals of the *foreground* (the sound of current interest) and the *background* (all other sounds in the mix). In this model, the *loudness* of the foreground signal is defined as the model-predicted human perception of the intensity of the sound in isolation. *Partial loudness* is the model's prediction of human perceived loudness of the foreground signal in the context of the background. Our implementation of the model depends on calculating loudness in individual frequency bins. We call the loudness of a certain frequency bin *specific loudness*. We call the partial loudness at that frequency bin *specific partial loudness*.

We now provide a brief overview of the GM model. For more detail please see the original paper [3]. For a practical overview of how to implement the GM model, see [20].

3.1. Overview of Glasberg and Moore's Time-varying Model of Loudness and Partial Loudness

The GM model approximates the transformations that occur between the time a sound pressure wave reaches the ear and when it is perceived by the brain. There are three important stages in the human auditory system that the GM model accounts for: outer/middle ear, basilar membrane, and cochlear hair cells firing signals to the brain.

Stage 1: The first stage of the model is to approximate the transformations that take place in the outer/middle ear. Each track is passed through an experimentally-determined transfer function (implemented as a 4097 coefficient FIR filter) that models the frequency response of the sound pressure transmission through the outer and middle ear towards the cochlea.

Stage 2: The second stage of the model approximates the response of the basilar membrane, which is the membrane within the cochlea that vibrates at different locations depending on the strength of various frequencies of an input sound. This stage approximates the basilar membrane motion by modeling its excitation (the intensity with which it is vibrating) at various points given the input to the inner ear.

This is done as follows: Let x_m be the time-series of audio samples for a singe track m, after it has been passed through the first stage filter. For simplicity of notation, assume x_m has already been windowed to a single frame of analysis (e.g. a chunk of 1024 samples). A multi-resolution Short Time Fourier Transform (STFT) is performed on x_m. The output of this STFT is then integrated over equivalent rectangular bandwidth (ERB)-spaced frequency bands to simulate the excitation pattern of the basilar membrane. The center frequencies of these bands range from 50 Hz to 15 kHz. The widths of these frequency bands were experimentally determined so that each has an equal contribution to overall loudness. The output of this integration is called the *excitation pattern*. The excitation pattern for track m at filter center frequency f is denoted $E_m[f]$.

Stage 3: The final stage of the model approximates perceived loudness from the excitation pattern by modeling the response of cochlear hair cells. Each bin of the excitation pattern is transformed into a single *specific loudness* value $L_m[f]$. This models perceived loudness of the track in isolation. These values are calculated using a piece-wise function that imposes different compressive non-linearities depending on whether or not the excitation level is above or below the threshold of excitation [3]. The model also outputs *specific partial loudness* value $PL_{m,n}[f]$. The partial loudness can only be determined in context of other sound, as the threshold of excitation for target track m is modified to model masking in light of a context background track n.

Lastly, for time-varying sounds, the loudness and partial loudness are smoothed over time using conditional filter coefficients based on whether the sound is in an attack or release phase. This temporal integration models forwards and backwards temporal masking. Note that, in our current implementation of MixViz we do not implement the temporal masking in the model because it is subsumed in the smoothing function of our visualization, described in Section 4.

3.2. Extension of the GM Model by Ward et al.

In MixViz, we want to visualize how the loudness of sounds vary by frequency. We also want to detect masking in individual frequency bands. Specific loudness and specific partial loudness are well suited for this since they are already split up into bins by frequency. However, the GM model only takes two input audio signals whereas we need to calculate

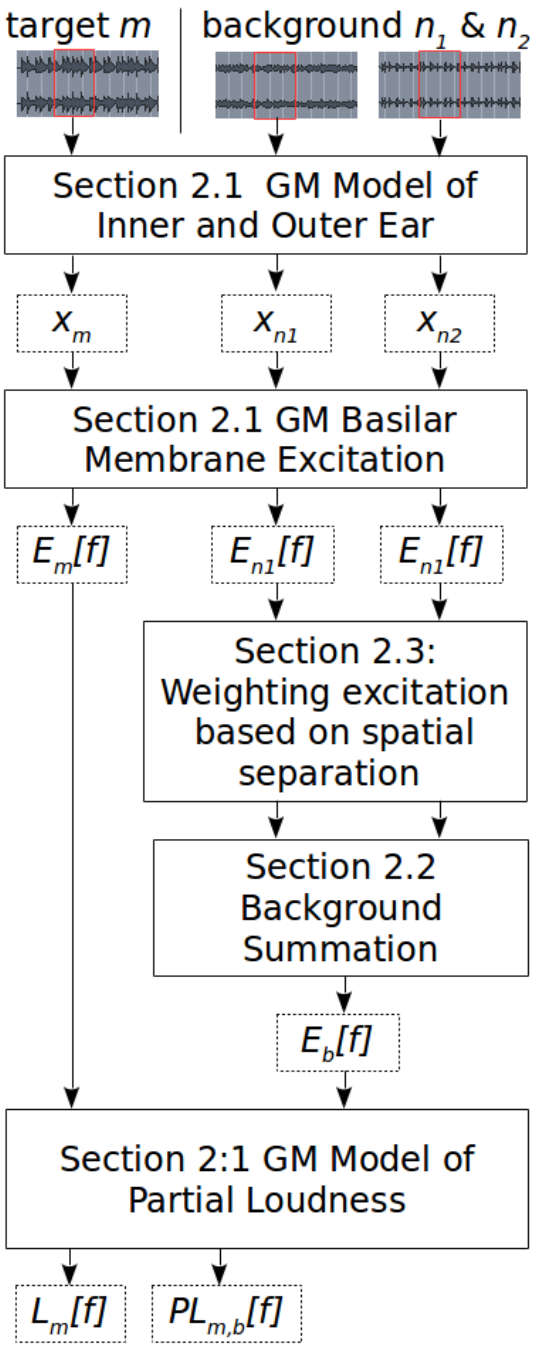

Fig. 1: Calculating specific loudness ($L_m[f]$) and specific partial loudness ($PL_{m,b}[f]$) for a single window/frame of target track m in a mix of two background tracks n_1 and n_2. Together n_1 and n_2 are called b, the background track. This process is repeated for all tracks.

masking and partial masking among several audio signals. To address this problem, we borrow the cross-adaptive model used in Ward's system for automatic mixing [19]. Ward's model treats the target track m as a single entity and sums the excitation patterns of all the other tracks n into a single background track.

$$E_b[f] = \sum_{n \neq m} E_n[f] \tag{1}$$

In Ward's work the specific partial loudness of the target track m is calculated via Stage 3 of the GM Model as discussed in Section 3.1. The process is repeated for each visualized target track in the mix.

3.3. Accounting for Spatial Release from Masking via Spatial Cues

Release from masking based on spatial cues is another issue not addressed in the GM model that is relevant to a mixing visualizer. The system of Ward et al. also does not take into account spatial release from masking. Therefore, we have extended this work to take spatial release into account.

As mentioned in Section 1, if two sounds are sufficiently spatially separated, then the human auditory system can use spatial cues to help separate the sounds and provide some amount of masking reduction. In a 2005 study by Marrone et al., values for the actual amount of masking reduction for different degrees of spatial separation were measured experimentally [2]. We incorporate this data in our model.

To infer the spatial position of a track in MixViz, we require stereo tracks so that we can infer spatial position for each track. This is distinct from existing systems, which use single-channel tracks. For a given track we calculate the angular position of each frequency bin $\theta[f]$ by simply comparing the intensity of the left and right channels and linearly mapping to a range of $-90^\circ - 90^\circ$ (all the way left to all the way right). While this is an oversimplification, it is a useful first approximation. Next, the spatial position θ of the track is calculated by taking the weighted average of the spatial positions (weighted by the magnitude of the value in that frequency bin, M). Here F is the number of frequency bins.

$$\theta = \frac{\sum_{f=1}^{F} M[f]\theta[f]}{\sum_{f=1}^{F} M[f]} \tag{2}$$

After calculating the spatial position θ_n of each track n, we can find the spatial separation $S_{m,n}$ of two tracks m and n by taking the absolute value of their difference. Thus, the spatial separation between all pairs of tracks is known.

When calculating the partial loudness of a track m (the target track), the intensity of all other tracks (the background tracks) must be adjusted according to their spatial separation from the target track. We do this to account for the release from masking via spatial cues.

We have created a formula for scaling spatial release from masking (noted as R, measured in decibels) as a function of angular distance between two sound sources m and n based on data reported by [2]. Specifically, we used the data reported for the reversed-speech condition in a reverberant room. The formula is as follows:

$$R_{m,n}(S_{m,n}) = -7.974 * (1 - e^{-0.103 * S_{m,n}}) \quad (3)$$

We then combine all background tracks into a single track, in a manner similar to Ward et al., but scaled by the spatial release from masking. Since the excitation pattern is not in terms of decibels, we exponentiate to return to a non-decibel format.

$$E_b[f] = \sum_{n \neq m} 10^{\frac{R_{m,n}}{20}} E_n[f] \quad (4)$$

Thus, spatial release for masking is accounted for in MixViz when summing background tracks to create the background excitation pattern E_b.

Now these excitation patterns are handed to Stage 3 of the GM model in a manner identical to that done by Ward. The output of this stage returns the loudness $L_m[f]$ of each track when taken in isolation, as well as partial loudness when taking the background into account, $PL_m[f]$.

Again, if the difference between specific loudness at a frequency f and specific partial loudness at f is above a certain threshold, then that frequency bin is considered masked. Therefore, frequency bin f for track m is considered masked if the specific loudness exceeds the spatially-adjusted partial loudness by more than the threshold T. T is user-adjustable, which is furthered described in Section 4.

$$(L_m[f] - PL_m[f]) \leq T \quad (5)$$

To implement this model in real-time, we extended Ward's loudness library found at:
`https://github.com/deeuu/loudness`

To do this, we added capability for the library to accept an arbitrary number of tracks at once as well the ability to calculate partial loudness. In addition, we implemented our extension to account for release from masking due to spatial separation. Our library can be found at:
`http://dx.doi.org/10.5281/zenodo.21966`

4. THE VISUALIZATION AND USER INTERFACE

After we calculate which tracks are masked in which frequency regions, we must convey that information to the user. MixViz presents the user with a 2-dimensional window that changes throughout time depending on what is happening in the audio mix. The horizontal dimension is spatial location (left to right) and the vertical dimension is frequency. Notice that these two dimensions correspond to the two methods mentioned earlier that users can use to reduce masking in a mix.

This visual representation of the mixing space is designed to let the user intuitively see how to reduce masking in their mix (for further explanation, see Section 5). The MixViz interface is displayed in Figure 2 with some annotations. We will now provide an overview of how the output of our custom GM model is mapped onto the visualization.

The three parameters in the HSV (hue-saturation-value) color space map particularly well to values calculated in our masking model. Each track is a assigned a color hue. Hues are equally spaced to minimize visual confusion between different tracks. The value (sometimes called brightness) of a point is determined by the specific loudness of the track at a certain frequency bin, as calculated by the masking model. Thus, the strongest frequency regions of a track appear as the most intense colors in the visualization, and weak frequency regions appear as near black (see strong and weak regions of Figure 2). The background of the visualization is black, or value 0. If a frequency bin has been labeled as masked, then it is drawn with a high saturation, creating a lightened (more white) color, as shown in the masked region of Figure 2. When colors/tracks overlap, only one color is shown. We chose to not perform any color

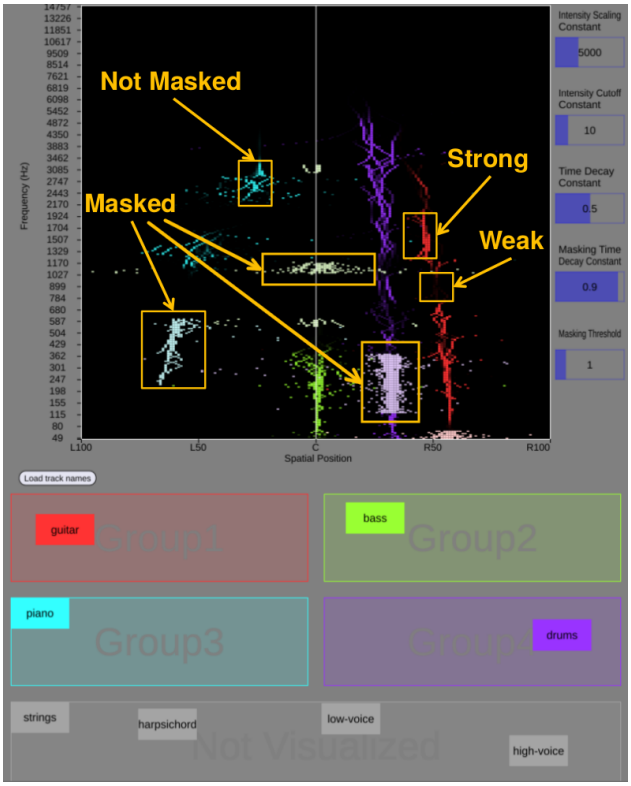

Fig. 2: A single frame of the MixViz interface with some annotations. The rendered visualization represents one time window of audio.

mixing because it is possible that two colors could mix to become the color of another track, which could cause confusion between the two conflicting tracks with a mixed color and whichever track happens to be regularly visualized with that mixed color. In addition, we believe that it is obvious when the colors from two tracks are competing for the same region of the visualization because the visualization is transient enough such that both colors will be displayed in the region when viewed over many frames.

There are several user-adjustable parameters that are factored in when rendering the visualization. The *intensity constant* is a constant that is divided by the specific loudness of a frequency bin to calculate the value (from hue-saturation-value) when drawing that bin. Thus, a higher intensity constant essentially decreases the sensitivity of the visualization to loudness and causes a color that would have been more intense to be less intense.

The *intensity threshold* determines the minimum specific loudness that a sound must have to be visualized at all. This lets the user adjust the visualization to exclude room tone or background noise. The *masking threshold* determines the minimum threshold magnitude difference between the specific loudness and spatially adjusted specific partial loudness of a frequency bin for it to be considered masked. It controls parameter T in equation 5.

The *time-decay constant* ranges from 0 to 0.99 and represents the amount of time smoothing that will take place in the visualization. In effect, this replaces the time-smoothing in the GM model. We chose to do this because the time smoothing constants used in the model resulted in visualizations that were too transient for adequate visualization of masking. We therefore decided to put the smoothing window size in the hands of the user. At a value of 0, transients will quickly appear and then disappear; at a value of 1, transients will remain on the screen for several seconds. The exact duration depends on a transient's amplitude.

All of visualization parameters are set to default values that should be acceptable for most use cases when the visualizer is initialized.

Of course, there are limitations to how much visual information a user can process simultaneously. If too many tracks were rendered in MixViz simultaneously, the visualization could become cluttered and it would be difficult for the user to distinguish between different tracks. To combat this issue, we added the capability to group tracks together and treat them as a single source. One might think that this feature is in direct opposition to the high-level mixing goal that all tracks should be clear and distinct, but in practice, we find users prefer to work in groups of tracks. For example, there may be a string section in the song where it is not important that the individual instruments are clear, but it is important for the strings to be distinct from other tracks in the mix. The track grouping capability of MixViz lets the user easily click and drag the tracks they want grouped together to the same group box.

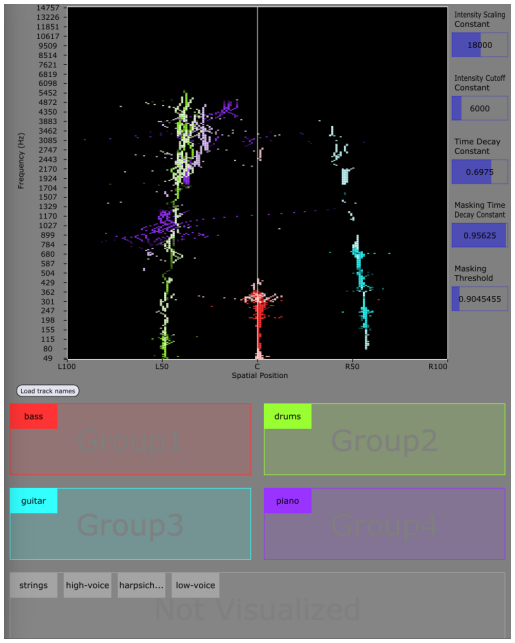

Fig. 3: In this MixViz frame, the piano and drums are masking each other.

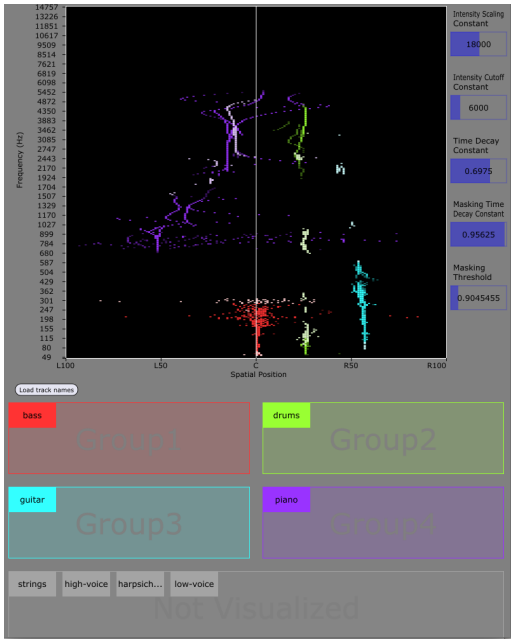

Fig. 4: The drums have been panned to the right since the user saw they were masked in their previous location in Figure 3. Now there is less masking (white) in the piano and drums.

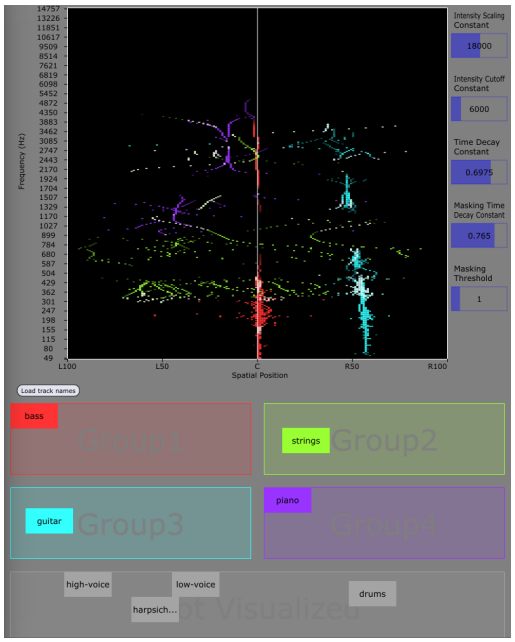

Fig. 5: In this MixViz frame, the strings (green) are masking the guitar (blue) at about 375-425Hz.

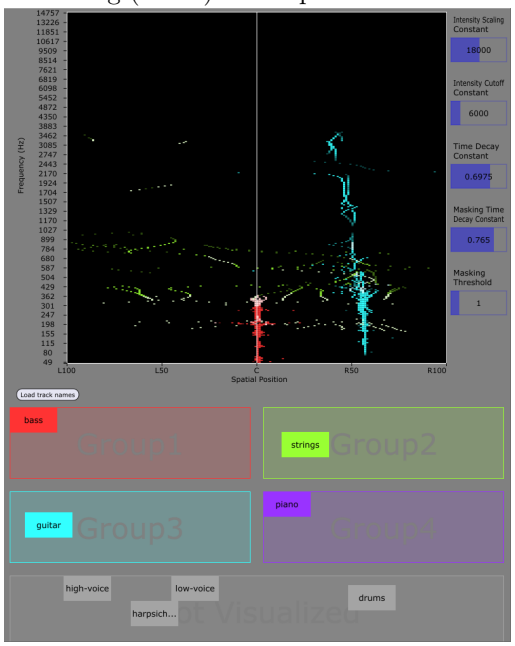

Fig. 6: The user has applied equalization to reduce 400Hz with $Q = 0.25$ by 4dB. Now there is significantly less masking (white) in the guitar in this frequency range.

5. EXAMPLE USE SCENARIO AND DESCRIPTION OF WORKFLOW

We will now describe an example scenario in which MixViz enables a user to detect and eliminate masking in an audio mix to achieve their mixing objectives. We hope that this scenario helps both to explain how to use MixViz and to illuminate the usefulness of the MixViz interface.

Consider the case where a user is mixing a song containing eight tracks. First, they want to make sure that the bass, drums, guitar, and piano are all clear and distinct in the mix. Figure 3, shows a screenshot of these four tracks in MixViz.

It can be seen from the visualization that the drums (green) and purple (piano) are interfering with each other and causing some masking (the whitish green and purple). The user sees that there is some open space on the right channel of their mix and pans the drums to the right to create some spatial separation between drums and piano. Figure 4 demonstrates that this action effectively eliminated a lot of the masking that was occuring. The piano and drums should now should more clear and distinct in the mix.

Next, the user wants to make sure that the guitar in their mix is clear and distinct from the strings. So, the user drags drums into the gray group (not visualized) and strings into the green group (group 2). Figure 5 shows that the guitar is currently masked.

It is clear from the visualization that the strings are the only track group with strong energy in the same frequency bands where the guitar is masked, so the user knows that the strings are causing this masking. In addition, the guitar was not masked before the drums were switched out with the strings track as can be seen in Figures 3 and 4. The user applies equalization to the strings to reduce 400Hz with $Q = 0.25$ by 4dB. Figure 6 shows the result of this action. It is clear from the visualization that the guitar is no longer significantly masked by the strings.

Thus, MixViz was used in multiple mixing scenarios to help reduce masking and create a better sounding mix.

6. FUTURE WORK

One limitation of MixViz is that its spatial location detection algorithm is relatively naive, simply comparing the magnitude of the left and right channels. When tracks are grouped together in MixViz, the group is displayed on the screen with many spatial positions and occupies a wider left-right distribution than is likely warranted.

Future work includes visualizing a third dimension, which could potentially be the "perceived depth" of a track in the mix. This could help users focus on the goal of manipulating foreground and background elements. To do this an accurate algorithm to calculate perceived depth of a sound would have to be developed.

Currently MixViz only provides visualization. Any panning or equalization must be done in the digital audio workstation connected to MixViz. In the future, we plan to add the ability to manipulate audio directly in MixViz. For example, panning a track by clicking and dragging it. We believe this would make MixViz an even more intuitive interface for creating a high quality audio mix.

7. CONCLUSION

When mixing many sounds together in an audio mix, tracks begin to interfere with each other and cause masking that muddies the mix. In many multi-track mixtures it is difficult for non-experts to determine which tracks are masked and in which frequency ranges. We have created MixViz, a tool that identifies which tracks and masked and in which frequency ranges. In the process, we extended the Glasberg and Moore Model of Time-Varying Loudness and Partial Loudness to account for spatial release from masking effects. MixViz should both lessen the learning curve for novices and improve the efficiency of experts when creating a high quality audio mix.

8. ACKNOWLEDGEMENTS

This work was funded, in part, by the United States National Science Foundation awards 1420971 and 1116384.

9. REFERENCES

[1] B. C. Moore, *An introduction to the psychology of hearing*. Brill, 2012.

[2] N. Marrone, C. R. Mason, and G. Kidd Jr, ''Tuning in the spatial dimension: Evidence from a masked speech identification task,'' *The Journal of the Acoustical Society of America*, vol. 124, no. 2, pp. 1146--1158, 2008.

[3] B. Glasberg and B. Moore, ''A model of loudness applicable to time-varying sounds,'' *Journal of the Audio Engineering Society*, vol. 50, no. 5, pp. 331--342, 2002.

[4] A. Sabin, Z. Rafii, and B. Pardo, ''Weighting-function-based rapid mapping of descriptors to audio processing parameters,'' *Journal of the Audio Engineering Society*, vol. 59, no. 6, pp. 419--430, 2011.

[5] A. T. Sabin and B. Pardo, ''2deq: an intuitive audio equalizer,'' in *Proceeding of the seventh ACM conference on Creativity and cognition*, Series 2DEQ: an intuitive audio equalizer, (1640339), pp. 435--436, ACM, 2009 Published.

[6] D. Reed, ''Capturing perceptual expertise: a sound equalization expert system,'' *Knowledge-Based Systems*, vol. 14, no. 12, pp. 111--118, 2001.

[7] S. Mecklenburg and J. Loviscach, ''subjeqt: controlling an equalizer through subjective terms,'' in *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, Series subjEQt: controlling an equalizer through subjective terms, (1125661), pp. 1109--1114, ACM, 2006 Published.

[8] M. Cartwright, B. Pardo, and J. Reiss, ''Mixploration: Rethinking the audio mixer interface,'' in *International Conference on Intelligent User Interfaces*, Series Mixploration: Rethinking the Audio Mixer Interface, ACM, 2014 Published.

[9] P. Seetharaman and B. Pardo, ''Crowdsourcing a reverberation descriptor map,'' in *Proceedings of the ACM International Conference on Multimedia*, Series Crowdsourcing a Reverberation Descriptor Map, pp. 587--596, ACM, 2014 Published.

[10] R. Selfridge and J. D. Reiss, ''Interactive mixing using wii controller,'' in *130th Convention of the Audio Engineering Society*, Series Interactive mixing using Wii controller, Audio Engineering Society, 2011 Published.

[11] B. L. Schmidt, ''A natural language system for music,'' *Computer Music Journal*, vol. 11, no. 2, pp. 25--34, 1987.

[12] S. Heise, M. Hlatky, and J. Loviscach, ''Automatic adjustment of off-the-shelf reverberation effects,'' in *126th Convention of the Audio Engineering Society*, Series Automatic Adjustment of Off-the-Shelf Reverberation Effects, 2009 Published.

[13] H. Katayose, A. Yatsui, and M. Goto, ''A mix-down assistant interface with reuse of examples,'' in *International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution*, Series A mix-down assistant interface with reuse of examples, p. 8 pp., 2005 Published.

[14] M. J. Terrell, A. J. Simpson, and M. B. Sandler, ''A perceptual audio mixing device,'' in *134th Convention of the Audio Engineering Society*, Series A Perceptual Audio Mixing Device, Audio Engineering Society, 2013 Published.

[15] B. De Man and J. D. Reiss, ''A knowledge-engineered autonomous mixing system,'' in *135th Convention of the Audio Engineering Society*, Series A knowledge-engineered autonomous mixing system, Audio Engineering Society, 2013 Published.

[16] E. Perez-Gonzalez and J. Reiss, ''Automatic equalization of multichannel audio using cross-adaptive methods,'' in *127th Convention of the Audio Engineering Society*, Series Automatic Equalization of Multichannel Audio Using Cross-Adaptive Methods, Audio Engineering Society, 2009 Published.

[17] E. Perez-Gonzalez and J. Reiss, *Automatic Mixing*, pp. xxi, 602 p. Chichester, West Sussex, U.K.: Wiley, 2nd ed., 2011.

[18] J. Scott, M. Prockup, E. Schmidt, and Y. Kim, "Automatic multi-track mixing using linear dynamical systems," in *Sound and Music Computing*, Series Automatic Multi-Track Mixing Using Linear Dynamical Systems, 2011 Published.

[19] D. Ward, J. D. Reiss, and C. Athwal, "Multitrack mixing using a model of loudness and partial loudness," in *133rd Convention of the Audio Engineering Society*, Series Multitrack mixing using a model of loudness and partial loudness, Audio Engineering Society, 2012 Published.

[20] A. J. Simpson, M. J. Terrell, and J. D. Reiss, "A practical step-by-step guide to the time-varying loudness model of moore, glasberg, and baer (1997; 2002)," in *134th Convention of the Audio Engineering Society*, Series A Practical Step-by-Step Guide to the Time-Varying Loudness Model of Moore, Glasberg, and Baer (1997; 2002), Audio Engineering Society, 2013 Published.